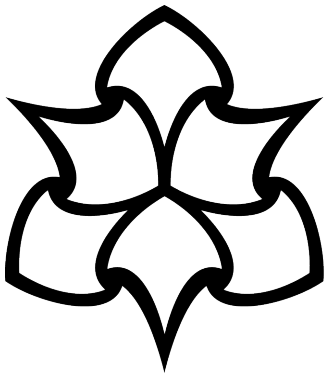

Automated Data Processing of Bank Statements for Cash Balance Forecasting

Abstract. The forecasting of cash inflows and outflows across multiple business operations plays an important role in the financial health of medium and large enterprises. Historically, this function was assigned to specialized treasury departments who projected future cash flows within different business units by processing available information on the expected performance of each business unit (e.g. sales, expenditures). We present an alternative forecasting approach which uses historical cash balance data collected from standard bank statements to systematically predict the future cash positions across different bank accounts. Our main contribution is on addressing challenges in data extraction, curation, and pre-processing, from sources such as digital bank statements. In addition, we report on the initial experiments in using both conventional and machine learning approaches to forecast cash balances. We report forecasting results on both univariate and multivariate, equally-spaced cash balances pertaining to a small, representative subset of bank accounts.

Keywords: Time Series Forecasting, Cash Flow Forecasting, Data Wrangling.

1 Introduction

Cash flow forecasting is a critical task for corporations of all sizes and across the whole spectrum of business activities. The more diverse these activities are, the more demanding it is for the company stakeholders to make informed financial decisions. Teams of experienced treasurers are involved in estimating the future available cash of corporations and making investment decisions based upon these estimations. Every hour spent on investigating the financial information results in a non-negligible opportunity cost to the business stakeholders, especially in volatile times when the distribution of resources needs to be closely monitored. The benefits of using analytical methods to issue cash flow forecasts based on historical data are, therefore, twofold. Firstly, there is the potential to improve the forecasting accuracy, which optimizes the resource allocation across the business and, secondly, the potential to improve the performance of the treasury departments, shifting the focus from low-yield data collection tasks to lucrative investment decision making.

This paper addresses the problem of forecasting daily cash balances of corporate bank accounts by modeling the data as equally-spaced time series. In line with the typical cash operations of medium and large enterprises, the forecasting time horizon of our predictive models has been set to one full month (22 business days). This requirement for predictions over larger time spans seems to be a distinguishing feature and additional challenge addressed in this work, when compared to other common

financial time series modelling tasks (e.g., stock prices movement). The data in this research is extracted from bank statements, which are collected through the SWIFT network (see Section 4.1). The information contained in bank statements is aimed at providing live cash visibility and transparency across bank accounts. These are a standardized form of communicating financial information, which means that a forecasting system that consumes bank statements can be used by a large number of companies. By focusing on reporting the overall liquidity within the account, bank statements often neglect reporting information at the transactional level, which impedes the flexibility of analytical forecasting methods. In our approach, we have identified and addressed some significant challenges (Section 4) in data collection and pre-processing, as well as with forecasting itself. Some of these challenges, such as inconsistency of transactional data reconciliation, irregularity of the statement issue time and missing data, are faced while reconstructing the cash balance data from the bank statements of different accounts. Other challenges revealed in the modeling process are related to the reduced historical data, operational outliers and pattern changes in cash balances. The main contribution of this paper is the assessment of these practical challenges and the proposal of solutions to alleviate them, with a view of scoping the prediction of cash balances based on bank statement data as a pure time series forecasting problem. By addressing the challenges, this paper exemplifies the use of both conventional (SARIMA, TES) and machine learning (ANN) models to issue cash balance forecasts in an automated and scalable manner. The scalability of the approach presented in this paper is compared to the historical (and still conventional) method used in cash flow forecasting. This method requires manual data aggregations and domain experts to estimate the expected performance of different business units within an organisation.

This paper is organized as follows: Relevant terminology is first introduced in Section 2, followed by a description of the data in Section 3. Section 4 describes data challenges associated with account balance forecasting. Section 5 presents related work on both conventional and machine learning methods applied to the problem of time series forecasting. Section 6 describes the experimental methodology used to compare the performance of several conventional and machine learning methods across cash balance accounts selected to illustrate the identified challenges. Section 7 presents the conclusions and future directions.

2 Terminology and Notation

A time series dataset is a series of values of one variable (univariate) or multiple variables (multivariate) that are organized in an ordered structure provided by the time component of the series. The time component of a time series not only enriches the series with information, but also sets constraints on the dependencies between the values of the variables in the series. In that respect, all entries of a time series are interdependent, which constraints the sampling methods that can be applied to the data.

A forecast of a time series is defined as an ordered prediction of the future values of the series. We define a time series y_k, where y_k takes values from the ordered group $y_1, y_2, \dots, y_n$. A forecast $F_{n+1}, F_{n+2}, \dots, F_{n+m}$ of the series is defined as a prediction the values of the series y_k, over the period of $n, n+m$ days, where m is the forecasting

horizon. The accuracy of a forecast is inferred from the deviation of the forecast from the actual values of the series over the forecasting horizon. There are multiple functions that can be used to compute the deviation. In this paper, we used the normalized root mean squared error (NRMSE) and the symmetric mean absolute percentage error (SMAPE) defined below:

$$NRMSE = \sqrt{\frac{\sum_{i=n+1}^{n+m}(F_i - y_i)^2}{m * \sum_{1}^{n}(y_i - \bar{y})^2}}, \quad (1)$$

$$SMAPE = \frac{100\%}{m} \sum_{i=n+1}^{n+m} \frac{|F_i - y_i|}{(|F_i| + |y_i|)/2}. \quad (2)$$

A transfer function f is used to map a subset of the past values of the series to the forecasted values. Depending on the forecasting methods used, the subset of past values can vary in size. A good way to choose the subset size is by analyzing the autocorrelation function of the series. The autocorrelation function is the correlation of the signal with a lagged copy of itself as a function of the lagging steps. A strong correlation for a certain number of lagged steps l indicates that the value of the k^{th} entry in the series has a strong influence on the $(k+1)^{th}$ entry, and therefore can be used to predict it. Similarly, a rapid drop in the autocorrelation function at lag l' indicates that the entries past the l'^{th} do not influence the prediction power of the transfer function, suggesting feeding the function a subset of l' series entries. So far, only the univariate time series forecasting problem has been discussed. A multivariate time series problem can contain, along the target series y_k, both endogenous and exogenous variables. An input variable is exogenous if it influences the target variable without being influenced by it; and endogenous if it can be influenced by the target variable. For example, the day in the month might influence the corporate cash balance due to the seasonality of cash operations, however, the cash balance does not influence what day it is. In contrast, the largest daily transaction within a bank account influences and is influenced by the daily cash balance of the account.

3 Data Description

The format of the data collected for this work is standardized by the Society for Worldwide Interbank Financial Telecommunication (SWIFT), the provider of a global network used for financial transactions. Depending on the scope of the information transferred, SWIFT uses different messaging types (MT). The types referring to cash management are under the format MT9xx. The two messaging types used in this work are MT940 and MT942. MT940 is the format used for end-of-day bank account statements whereas MT942 is the format used for intraday reporting. An MT940 statement includes the list of transactions having cleared during a business day and a MT942 statement includes a subset of the transactions cleared from the previous statement onwards. Although the standard of the messaging types for cash management

services is ensured by SWIFT, different banking entities have different data submission conventions maintained within their various legacy systems.

The bank statements issued for corporate users are similar in format to the retail bank statements. They contain a header detailing the account name, identification code, datetime of issue and the opening balance and closing balance. The bulk statement contains the list of transactions that sum up to the difference between the closing and the opening balance. Each transaction has an entry date, value date, amount, and several optional references: funds code, transaction type, identification code, reference to account owner and information for account owner. A subset of relevant features synthetized from a statement is: "'datetime': '2020-01-30 21:30', 'open': £400000, 'close': £387000, 'transactions': {'value date': '2020-01-31 00:00', 'value':+£4000, 'identification code': 'CHK', 'ref':'12354538 Cheque Company A', etc.}".

This section outlined the richness of the information contained in a standard bank statement. Section IV will discuss the specific challenges associated with the extraction of consistent and ordered cash balances from various bank statements.

4 Account Balance Data Challenges

The challenges presented by time series forecasting are well known. Fine-tuned models developed through conventional or machine learning methods suffer from time instability due to the lack of stationarity, degrees of uncertainty in historical time periods and the restrictions on train, test and validation splits due to time sequences, see e.g. [1]. Time Series data corresponding to financial transactions, however, pose additional difficulties, which do not seem to have been addressed in the academic literature yet.

4.1 Reconstruction Challenges

Inconsistency of Transactional Data Reconciliation. Some challenges of integrating merging financial time series data into a consistent format were discussed in [2]. Due to the global nature of many corporations, international payments and markets have an effect on statement data. Statements received from the bank are given relative timestamps based on time zones, meaning that collating transactions from multiple regions can lead to discrepancies. One such discrepancy often occurs when aggregating intraday statements (MT942) and reconciling against end of day statements (MT940), due to transactions having value dates past the issue date of the end of day statements.

Irregularity in Statement Time of Issue. In addition to reconciling transactional information within bank statements at the global market scale, a forecasting system needs to consider the temporal component of the cash balance time series obtained from bank statements. The temporal component of a statement of a bank account can be inferred as the time of issue of the statement by the banking institution managing the account. Because different banking institutions have different legacy systems for collating transactional information into statements, the statement reporting offering

varies across regions and legislations but also across banks within the same country. Most affected by this inconsistency are the intraday statements which need to be assigned an accurate date and time to allow for the reconstruction of the corresponding time series. With the statements arriving at different times during the day, only an intermittent time series can be reconstructed. Additionally, the average number of statements collected per day is between 2 to 5, limiting the potential of resampling methods as a solution to the series intermittence. Given the circumstances, we chose to de-scope the use of intraday statements from the current study.

Missing Data. Notwithstanding the irregularity in time of issue, assuming that there are a small number of transactions at the time when the end of day statement is collated (usually mid-night), the reconstructed time series of the end of day balances should be continuous and equally spaced. However, there are other factors that influence the data collection process, within a live environment. For example, there can be a temporal downtime of a service yielding interruptions or duplications of data that needs to be investigated manually. A specific example is the closing available balance which might be reported only sparsely or not reported at all for some bank accounts.

It is important to note that in other contexts that data seemingly missing at random can have a hidden serious bias (see e.g. [3]). Here, much of the data is missing purely due to different financial reporting conventions associated with the different accounts. Rather than being a specific problem with missing data per se, it is possible that there may be some hidden structure in the data associated with an intra-monthly effect identified by practitioners. This is something we wish to explore in a continuous-time model in future research [4].

4.2 Forecasting Challenges

Reduced Historical Data - Many of the state-of-art linear forecasting algorithms of univariate time series are based upon the extraction of seasonal patterns. These patterns are either extracted directly, in the case of stationary seasonality, or are indirectly trained upon through feeding the algorithms additional exogenous features of the time component (e.g. day of month, month of year, etc.). However, when the time series has less than one full period of a season, little information can be inferred on the seasonal pattern of the series. In the context of this study, many of the corporate accounts considered were recorded for between 6 to 9 months, which posed a challenge in modelling any potential pattern manifested for longer than a quarter. Additionally, depending on the commercial agreement between the client and their banking partners, the cash balances of the client's bank accounts are reported with different granularities. Some accounts are reported on every calendar day, accounts are only reported on working days, and others are reported at different days during the month. For example, there are accounts which are set to report only on the first Wednesday, Thursday and Friday of each month (Figure 1). The resulting time series of cash balances is sparse, with the balance information being populated at irregular time intervals that depend on the day of week rather than the calendar day. In both these situations, we chose to put the accounts with insufficient data points out of scope of this paper. Consequently, we excluded the bank accounts for which the reconstructed cash balance contained less than 15 data points per month (missing data) or less than 6 months' worth of data.

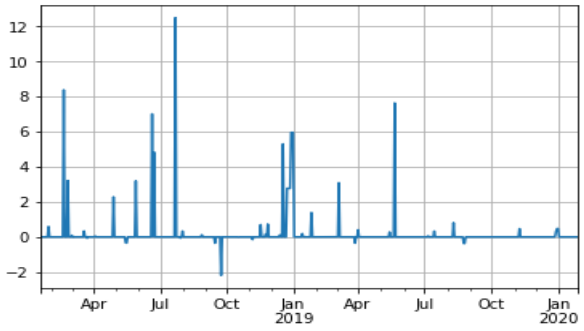

Fig. 1. Sparsely reported account. The y axis is a scaled true cash flow of the account.

Operational Outliers - Barring the stochastic component of the cash inflows and outflows, which are guided by external factors such as market volatility or client performance, the data contained within the statements issued for a business bank account is, a genuine representation of the cyclical business activities managed by the account. However, within the treasury management sector, it is often the case that seemingly random interventions of the domain specialists blurs away the valuable insights that the data has to offer (e.g. Figure 2). Operations such as inter-subsidiary lending, long term investments, mergers or acquisitions are traced in the cash balance as irregular sparks or dips which are challenging to be picked up through univariate modelling of the time series. The immediate solution to the challenges posed by the operational outliers is to flag out and eliminate the transactions corresponding to these outliers from the reconstructed cash balance and only model those transactions that are inherent to the business operations performed through the account. However, due to the subjective nature of the treasury management decisions, flagging these manual interventions has proven to be challenging even for domain experts that are external to the company department making the treasury decisions. Applying tests based on statistical metrics such as the standard deviation did not yield an improvement in forecasting performance. In the future we plan to use more sophisticated metrics for identifying the operational outliers, including smoothing functions and designated anomaly detection models.

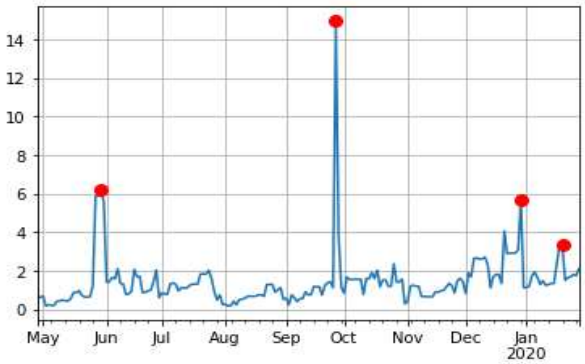

Fig. 2. Operational outliers revealed in an account sample. The y axis is a scaled true cash balance of the account.

Pattern Changes in Cash Balances - Large enterprise clients managing multiple production lines, brands and businesses present an additional layer of complexity into their cash balance sheets. Due to the volatility of the different markets their products address, decisions of cash allocations are being made at an increased pace, resulting in bank accounts being opened, closed or put out of use at various times during the year. One example of a cash balance with pattern change is shown in Figure 3. In this instance, the average cash balance suddenly drops during the period April to October 2019 with no respective behavior for the same period of 2018. Similar anomalies in the cash balances were considered separately by applying transformations to the time series and optimizing the model hyper-parameters in a manual manner.

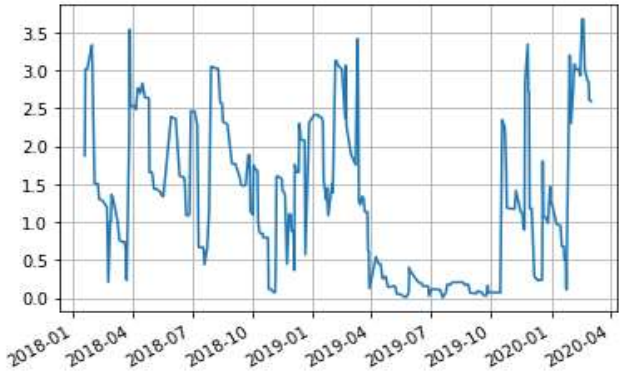

Fig. 3. Account sample demonstrating a pattern change in the period April - October 2019. The y axis represents the scaled true cash balance of the account.

5 Data Description

This section provides an overview of related work using both conventional and machine learning methods for financial forecasting. Traditional methods include Naïve Forecasting, which uses the actual cash flow data from a previous time period as the forecast for the upcoming period (REF). Simple Moving Average Forecasting (SMAF) adds the recent closing prices, then divides the total by the number of time periods to calculate the average. Exponential Smoothing uses exponential functions to assign exponentially decreasing weights over time periods and is useful when the more recent past is likely to have more impact on predicting a forecast than more historical past. For large historical cash balance datasets, autoregressive moving average (ARMA) and autoregressive integrated moving averages models (ARIMA) can be used to identify autocorrelations and are more suitable for long term forecasting. However, due to the challenges identified in section 4, there is no one size fits all solution.

5.1 Conventional Approaches to Financial Forecasting

It is important to recognize that finance presents idiosyncratic forecasting challenges that are significant in their own right. The inherently stochastic nature of the subject is further exacerbated by additional sources of short-term randomness that are typically modelled using an unobserved stochastic volatility component. Conventional forecasting approaches such as ARIMA are also typically at odds with notions of market efficiency [5], theoretical options-pricing models constructed via foundational arguments such as absence of arbitrage [6] and the stylized empirical facts of financial time series [7]. Financial time series modelling is also an inherently specialist area. Commonly used ARCH/GARCH models for (financial) time series are equivalent to ARMA models for an unobserved volatility component. However, a range of different model variations are possible [8]. This sheer range of available models underscores the specialist nature of the subject. Further extensions of these classical models have been used in applications that variously account for further autocorrelations in the observed series [9] and for additional regime-switching effects [10]. However, within this, the need to account for unobserved volatility fluctuations remains paramount.

Other non-time-series approaches to forecasting, e.g. those based around the technical analysis methods popularized by practitioners (see e.g. [9]), are possible. The academic literature on technical analysis is voluminous, see e.g. [12] or [13] for a review. However, such approaches have yet to permeate mainstream finance. An exception is [14] who use localized regression approaches to gauge the plausibility of technical analysis strategies. Thus, this serves to motivate the study of machine-learning techniques within financial forecasting. As an illustration, [15] reviewed corporate cash flow forecasting using account receivable data collected through a specialized accounting software, which provides a richer view of the individual transactions.

5.2 Machine Learning Approaches to Financial Forecasting

The use of deep learning for time series prediction, in specific domains is not new, but remains challenging due to the need for extremely large datasets of high quality data and the lack of transparency in how decisions were made. One of the most targeted areas is stock market forecasting predicting stock prices in different time slice windows. [16] created a deep learning framework which combined wavelet transforms, stacked auto-encoders and long-short term memory (LSTM) networks to predict six stock indices, one-step-ahead of the closing price. [17] utilized LSTM networks for predicting out-of-sample directional movements for a number of financial stocks and outperform other methods such as random forests. [18] proposed day-ahead multi-step load forecasting using both recurrent neural networks (RNN) and convolutional neural networks (CNN). In their work the use of the CNN model improved the forecasting accuracy by 22.6% compared to the application of seasonal ARIMAX. However, the dataset used was concerned with predicting accurate building-level energy load forecasts which looked at how similarities within data space can be identified in financial forecasting.

Whilst Deep Learning has been successfully applied in many domains, it is not always successful. Small datasets do not tend to perform well, with research indicating that to be successful, millions of data points are required. Data quality is always an issue when applying machine learning, the generalization error of artificial neural networks can be

improved by the addition of noise in the training phase. Consequently, this provides a barrier to be overcome with respect to forecasting financial time series. This may help to explain some of the data challenges described in section 3. Despite some recent progress, explaining and interpreting models remains challenging. Ensuring the financial interpretability of the deep learning models constructed is thus far from being a foregone conclusion.

Ensemble machine learning models have also been used for financial forecasting in e.g. [19] and [20]. [19] combined two traditional ensemble machine learning algorithms: random subspace and MultiBoosting to create a method known as RSMultiBoosting to try and improve the accuracy of forecasting the credit risk of small-to-medium companies. RSMultiBoosting outperformed traditional machine learning algorithms on small datasets and the ability to rank features according to the decision tree relative importance score improved accuracy. [21] conducted a study investigating several models including deep and recurrent neural networks and the CART regression forest to examine non-linear relationships between input and output features on abnormal stock returns generated from earnings announcements based on financial statement data. The results indicated that non-linear methods could predict the direction of the "absolute magnitude of the market reaction to earnings announcements correctly in 53% to 59% of the cases on average." [21] with random forest approaches providing the best results. Whilst this is a reasonable result, it highlights the issues of data quality (as discussed in section 2) and its impact on whether an account is forecastable.

6 Experimental Comparison

This section provides a comparative analysis of the performance of different time series forecasting techniques on a subset of anonymized time series that replicate real bank account cash balances. The sampling and pre-processing procedures are explained in subsection A. We report a novel approach of enriching the univariate time series pertaining to cash balance data by aggregating the transactions pertaining to bank statements by various statistical metrics (e.g. standard deviation). Subsection B describes the forecasting methodologies, beginning with the univariate approaches (SARIMA and TES), then enriching SARIMA with multivariate exogeneous constraints, and ultimately leveraging the multivariate input via a neural network architecture.

6.1 Dataset Description

To share the learnings gained from forecasting cash balance in various bank accounts, a sample time series dataset representative of cash balance data was created. The dataset consists of collections of time series corresponding to daily cash balances and some additional exogenous and endogenous variables. As described throughout the section on the data challenges, there are multiple granularities in which the account balance statements are recorded. The examples selected for this work are those for which the reconstructed time series contains at least one data point per business day. An additional level of complexity is given by the possibility of some accounts to consist of a group of individual bank accounts.

A sparse time series is created by collating the closing balance amounts with corresponding value dates. To provide a consistent view over the balances of multiple accounts of a client, the closing balances are converted to a currency of choice. The missing values in the sparse time series are then forward filled to the granularity set for the account (per business day or daily). The reason for filling the values with previous valid entries (forward filling) is that it is presumed that in the valid dates when the balance is not reporting, there were no cash movements. In the case of groups of accounts, the date range is initially established as the minimum and maximum dates reported by any of the bank accounts in the group. The missing values in each individual bank account are then filled, initially forward and then backward as well, to cover the period between the earliest statements of the group and the individual earlies statement. Subsequently, the individual continuous time series are summed up to the grouped time series. The exogeneous variables obtained from the time component of each time series are then computed. Endogenous variables obtained through applying various statistical aggregations to the transactional statement data are also used to predict the cash balance. However, due to potential clashes with the IP of AccessPay, the aggregation methods will not be discussed explicitly. The sample dataset used throughout the paper represents a selection of 6 bank account aggregates with end of day cash balances reported in each business day during the period 18 June 2019 – 27 January 2020. Figure 4 below shows each time series collected. A train-test split was applied, where the size of the test set is equal to the forecasting horizon of one month.

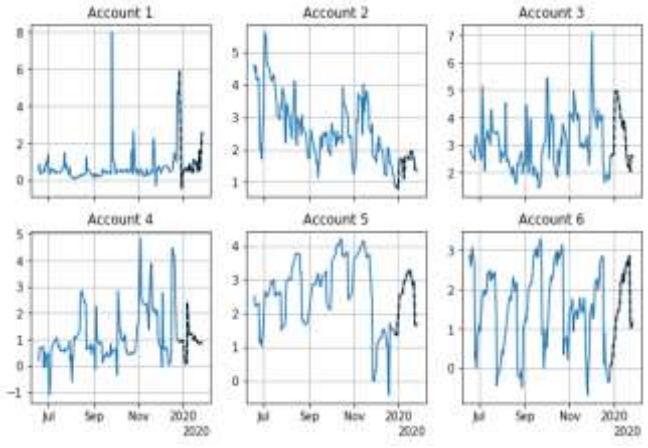

Fig. 4. Cash Balance Series of the Selected Accounts. The train set is shown in blue and the test set is shown in black. The y axis represents the scaled true cash balance of each account.

6.2 Experimental Results

The results of the forecasting experiments are separated in three sections. The first section discusses the problem of univariate time series forecasting via conventional methods, ARIMA and TES. In the second section we propose an extension of the ARIMA to account for multivariate input. The third section discusses the results of a machine learning approach based upon ANNs.

Conventional Univariate Time Series Forecasting. The conventional time series forecasting algorithms referred throughout this paper are the Seasonal Autoregressive Integrated Moving Average (SARIMA) and the Triple Exponential Smoothing (TES). The SARIMA model has seven hyper-parameters, one for the seasonal differentiation term, three for modelling the seasonal component of the series and three for modelling the remainder of the series. In the experiments undertaken for this paper, these parameters were determined through experimentation against the AIC score, leading to the SARIMA(1,0,1)(1,1,1)22. The value of 22 for the seasonal differentiation was chosen as the main number of business days in a month.

The Triple Exponential Smoothing features four hyper-parameters. These are the trend type, the damping of the trend, the seasonal type and the seasonal differentiation term. Based upon heuristics, it was found that the best set of values for these parameters are: trend: additive, damped: False, seasonal: additive and seasonal differentiation: 22.

The first three rows in Table 1 detail the performance of the two methods over the account samples, as compared to the Naive Average benchmark. While there are some accounts for which the TES prevails both the benchmark and the SARIMA model, overall, only SARIMA overcomes the benchmark in a consistent manner.

Table 1. Aggregate accuracy metrics on the account samples

Account	A1	A2	A3	A4	A5	A6	Mean
Root Mean Squared Error							
Naïve Mean	1.27	4.47	1.12	0.47	0.66	0.85	**1.47**
SARIMA	0.67	0.73	0.93	4.25	0.70	0.42	**1.28**
TES	5.51	0.3	8.66	2.22	0.17	1.05	**2.99**
MULTI SARIMA	0.67	0.56	0.62	4.67	1.55	0.53	**1.43**
ANN	1.42	3.12	0.44	1.38	1.33	0.43	**1.35**
Symmetric mean absolute percentage error							
Naïve Mean	53.17	60.38	26.86	33.98	22.84	50.87	**41.35**
SARIMA	64.44	26.68	24.37	61.79	29.59	51.91	**43.13**
TES	129.2	24.37	132.0	73.78	10.82	81.01	**75.24**
MULTI SARIMA	63.64	27.73	22.68	60.51	47.26	55.19	**46.17**
ANN	59.02	52.71	16.65	74.60	31.22	40.93	**45.87**

To understand the gains and failures of the conventional univariate time series forecasting methods, we looked at the extreme cases for which the methods either outperformed or underperformed the benchmark by a considerable margin. From Table 1, these are account 2 (Figure 5) and account 3 (Figure 6). On the second account sample, both ARIMA and TES yielded accuracies exceeding the benchmark as measured by both the NRMSE and the SMAPE metrics. Noticeably, the NRMSE score of TES was the global minimum across all methods and accounts tested.

A different outcome was observed for Account 3. In this example, the vague monthly seasonality is only captured by the ARIMA method while TES seems to be tricked by the outlier around December into predicting a decreasing trend across January. To conclude, the univariate models are unstable and fail to generalize on the multitude of

cash balance series. While some level of progress is achieved for a subset of accounts through these methods, they would ultimately be overwhelmed by the unresolved challenges discussed in Section 2, in particular the Operational Outliers. As these outliers appear into the cash balances as a result of the institutional decisions that are made based upon transactional data, it is hoped that through making use of the information contained within the transactional data more insights could be drawn to support the unidimensional forecasting.

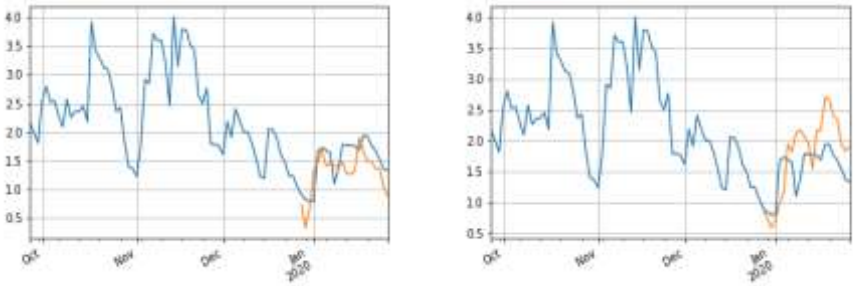

Fig. 5. Account 2 – Conventional forecasting outperformed the benchmark. The y axis represents the scaled true cash balance of the account.

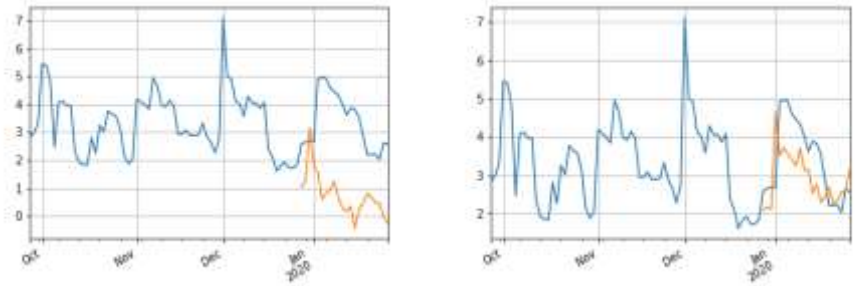

Fig. 6. Account 3 – Only SARIMA outperformed the benchmark. The y axis represents the scaled true cash balance of the account.

Conventional Multivariate Time Series Forecasting. As detailed in the dataset description (Section 6.1), a multivariate dataset was obtained through extracting information from transactional data contained within the bank statements. A series of aggregation techniques were applied on the transactional data, each based upon the different types of transactions and correlations between them. The aggregates obtained in a form of sparse time series are forward-filled to ensure there is no effect of future values on the past entries. As the sum of the transactions within a day reconcile the end of day cash balance, the aggregate transactional time series represent a set of endogenous variables to the target variable. Therefore, these series cannot be directly used as exogenous constraints to the SARIMA model. The solution implemented in this paper was to shift the time component of the transactional aggregates forward by the size of the test set. In other words, for example, the aggregates computed for the July

cash balances were used as exogenous constraints for predicting the cash balance during August.

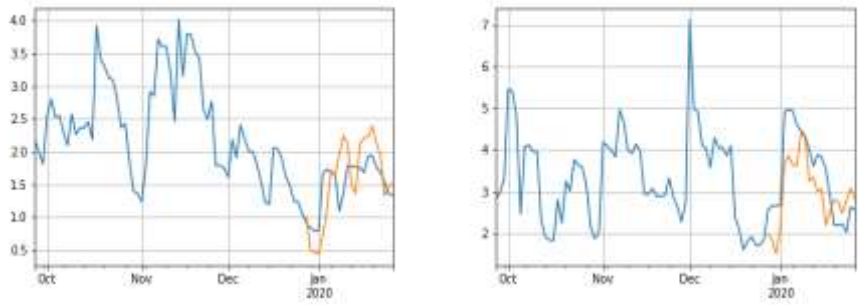

Fig. 7. Performance of SARIMA with transactional exogenous variables on accounts 2 and 3. The y axis represents the scaled true cash balance of each account.

Machine Learning Approaches. The improvement in accuracy through the use of exogenous constraints based upon past transactional data indicates that the transactional data could be leveraged to train endogenous regressors on top of the univariate signal from the cash balance. A family of models that were shown to be able to map multivariate inputs to time ordered outputs are the neural networks (e.g. [22]). The assumption that the transactional aggregates retain information about future cash balances, a stacked ensemble of a dimensionality reduction algorithm and an artificial network architecture was built. Due to the commercial nature of the experiment reported in this paper, the exact architecture of the neural network (ANN) cannot be revealed.

We report that the ANN architecture outperforms SARIMA with transactional exogenous constraints judged by both metrics used in this experiment. Compared against the univariate SARIMA, the machine learning method still underperforms, albeit by a small margin of 0.07 in the normalized root mean squared error metric. By comparing the individual performance over each account sample we can get a clearer picture of the difference between conventional and machine learning models.

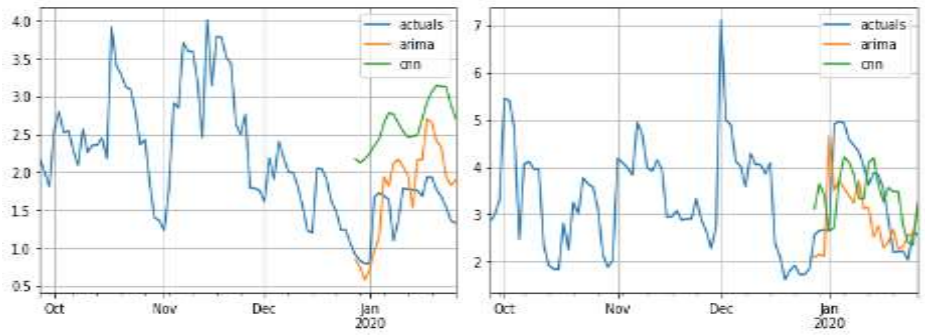

Fig. 8. Performance comparison of the univariate ARIMA and the multivariate NN methods on account 2 (0.73 vs 3.11) and account 3 (0.93 vs 0.44). The y axis represents the scaled true cash balance of each account.

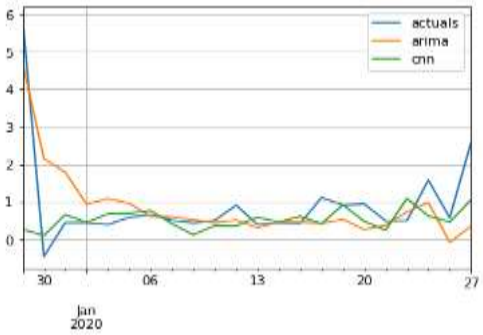

Fig. 9. Performance comparison of univariate ARIMA and multivariate NN over Account 1 (0.67 vs 1.42). The y axis represents the scaled true cash balance of the account.

Figure 8 presents a comparison of the accuracy of the two best performing models, univariate ARIMA and multivariate ANN, over the same two account samples discussed in previous sections. The 3 month historical actuals of the cash balances are included to give a view of the most recent trend in the series. Noticeably, while the ANN method outperforms by a large margin ARIMA on the Account 3, the performance over the second account is poor. We attributed the underperformance to the changing trend of the second account, which the ANN could not capture. In Figure 9 we compare the two models over a different bank account which contains a salient operational outlier. Similar to the case of the second account, the neural networks could not learn the rapidly changing pattern of Account 1. However, when eliminating the very first datapoint in the series, the root mean squared error drops from 1.42 to 0.45, whereas the SARIMA model yields the same error of 0.66 (Table 1). These observations suggest that, given the rapidly changing trends within the series, the ANN architecture can learn the smoothly varying features better than the conventional methods.

7 Conclusions and Further Work

In this paper we presented an overview of the challenges of understanding, collating and exploring account balance data pertaining to private enterprises with a view of forecasting their future cash balances. We showed several techniques for addressing these challenges, which allowed us exemplify the use of both conventional and machine learning techniques for cash balance forecasting. Additionaly, we performed a comparative analysis of conventional and machine learning based time series forecasting models on a representative subset of bank accounts. The intermediate results portrayed a fair competition between Seasonal Auto Regressive Moving Average and Neural Network Architectures, indicative of the stochastic nature of enterprise account cash balances. Permanent collaboration with field experts, either internal or external (banks and customers), and with the architects of the legacy code used for parsing the SWIFT statements is the ultimate solution to alleviating a number of challenges discussed in this paper. In the future, we plan on exploring the transactional features in more depth to get an understanding of the way they infer the

future values of the end of day cash balance. Additionally, through an improved collaboration with the domain specialists, we aim at limiting the influence of the challenges emphasized in this paper.

Acknowledgements

The authors would like to express their gratitude to the two anonymous referees for the helpful and supportive comments received.

References

1. Giles, C.L., Lawrence, S. & Tsoi, A.C. Noisy Time Series Prediction using Recurrent Neural Networks and Grammatical Inference. Machine Learning 44, 161–183 (2001)
2. Katselas, D., Sidhu, B., Yu, C. Merging time-series Australian data across databases: challenges and solution. Accounting & Finance 56, 1071-1095
3. Shang, Y. (2019). Subgraph robustness of complex networks under attacks. IEEE Transactions on Systems, Man and Cybernetics: Systems 49 821-832
4. Fry, J., Griguta, V-M., Gerber, L., Slater-Petty, H. and Crockett, K. (2021) Stochastic modelling of corporate accounts. Preprint.
5. Fama, E.: Efficient capital markets: A review of theory and empirical work. Journal of Finance 25, 383-417 (1970).
6. Merton, R. C.: The theory of rational options pricing. Bell Journal of Economics and Management Science 4, 141-183 (1973).
7. Cont, R.: Empirical properties of asset returns: stylized facts and statistical issues. Quantitative Finance 1, 223-236 (2001).
8. Hentschel, L.: All in the family: Nesting symmetric and asymmetric GARCH models. Journal of Financial Economics 39, 71-104 (1995).
9. Katsiampa, P.: Volatility estimation for Bitcoin: A comparison of GARCH models. Economics Letters 158, 3-6 (2017).
10. Walid, C., Chaker, A., Masood, O. and Fry, J.: Stock market volatility and exchange rates in emerging countries: A Markov-state switching approach. Emerging Markets Review 12, 272-292 (2011).
11. Meyers, T. A. The technical analysis course, 4th edn. McHraw-Hill (2011).
12. Park, C-H. and Irwin, S. H.: What do we know about profitability of technical analysis? Journal of Economic Surveys 21 786-826 (2007).
13. Nazário, R. T. F., e Silva, J. L., Sobreiro, V. A. and Kimura, H.: A literature review of technical analysis on stock markets. Quarterly Review of Economics and Finance 66, 115-126 (2017).
14. Lo, A. W., Mamaysky, H., Wang, J.: Foundations of technical analysis: Computational algorithms, statistical inference and empirical investigation. Journal of Finance 55, 1705-1765 (2000).
15. Weytjens, H., Lohmann, E. & Kleinsteuber, M. Cash flow prediction: MLP and LSTM compared to ARIMA and Prophet. Electron Commer Res (2019).
16. Bao, W., J. Yue, and Y. Rao, A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one Vol. 12(7), (2017)

17. Fischer, T. C. Krauss. Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, Vol: 270(2), 654-669 (2018).
18. Cai, M., Pipattanasomporn, M. and Rahman, S., 2019. Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. Applied energy, 236, pp.1078-1088.
19. Zhu, Y., Zhou, L., Xie, C., Wang, G.J. and Nguyen, T.V., 2019. Forecasting SMEs' credit risk in supply chain finance with an enhanced hybrid ensemble machine learning approach. International Journal of Production Economics, 211, pp.22-33.
20. Salas-Molina, F., 2019. Fitting random cash management models to data. Computers & Operations Research, 106, pp.298-306.
21. Amel-Zadeh, Amir and Calliess, Jan-Peter and Kaiser, Daniel and Roberts, Stephen, Machine Learning-Based Financial Statement Analysis (January 15, 2020).
22. Akram, M., & El, C. (2016). Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks. International Journal Of Computer Applications, 143(11), 7-11.